COMPLEXITY-OPTIMAL AND CURVATURE-FREE FIRST-ORDER METHODS FOR FINDING STATIONARY POINTS OF COMPOSITE OPTIMIZATION PROBLEMS[*]

WEIWEI KONG[†]

Key words. nonconvex composite optimization, first-order accelerated gradient method, iteration complexity, inexact proximal point method, curvature-free, adaptive, optimal complexity

AMS subject classifications. 47J22, 65K10, 90C25, 90C26, 90C30, 90C60

Abstract. This paper develops and analyzes an accelerated proximal descent method for finding stationary points of nonconvex composite optimization problems. The objective function is of the form $f+h$ where h is a proper closed convex function, f is a differentiable function on the domain of h, and ∇f is Lipschitz continuous on the domain of h. The main advantage of this method is that it is "curvature-free" in the sense that it does not require knowledge of the Lipschitz constant of ∇f or of any global topological properties of f. It is shown that the proposed method can obtain a ρ-approximate stationary point with iteration complexity bounds that are optimal, up to logarithmic terms over ρ, in both the convex and nonconvex settings. Some discussion is also given about how the proposed method can be leveraged in other existing optimization frameworks, such as min-max smoothing and penalty frameworks for constrained programming, to create more specialized curvature-free methods. Finally, numerical experiments on a set of nonconvex quadratic semidefinite programming problems are given to support the practical viability of the method.

1. Introduction.

Consider the nonsmooth composite optimization problem

$$\phi_* = \min_{z\in\mathbb{R}^n} \{\phi(z) := f(z) + h(z)\} \tag{1.1}$$

where $h : \mathbb{R}^n \mapsto (\infty, \infty]$ is a proper closed convex function, f is a (possibly nonconvex) continuously differentiable function on an open set containing the domain of h (denoted as $\operatorname{dom} h$), and ∇f is Lipschitz continuous. It is well known that the above assumption on f implies the existence of positive scalars m and M such that

$$-\frac{m}{2}\|x-x'\|^2 \le f(x) - f(x') - \langle \nabla f(x'), x - x'\rangle \le \frac{M}{2}\|x-x'\|^2 \tag{1.2}$$

for every $x, x' \in \operatorname{dom} h$. The quantity (m, M) is often called a *curvature pair* of ϕ, and the first inequality of (1.2) is often called *weak-convexity* when $m > 0$.

Recently, there has been a surge of interest in developing efficient algorithms for finding stationary points of (1.1). While complexity-optimal algorithms exist for the case where both m and M are known, a *curvature-free* algorithm — one without knowledge of (m, M) — with optimal iteration complexity remains elusive.

Our goal in this paper to develop, analyze, and extend a curvature-free accelerated proximal descent (CF.APD) algorithm that obtains, up-to-logarithmic terms, an optimal iteration complexity when f is nonconvex and a *near* optimal complexity (e.g., up to logarithmic terms) when f is convex. A rough outline of the $(k+1)^{\text{th}}$ iteration of CF.APD is as follows:

[*] **Funding**: The work of this author has been supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Versions: v1.0 (May 25, 2022)

[†] Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, 37830. wwkong92@gmail.com

Iteration $(k+1)$**:**

(i) Given $\hat{m} \in \mathbb{R}_{++}$, find a *proximal descent* point $z_{k+1} \in \operatorname{dom} h$ in which there exists $\hat{u} \in \mathbb{R}^n$ satisfying[a]

$$\hat{u} \in \nabla f(z_{k+1}) + \partial \left(h + \hat{m}\|\cdot - z_k\|^2\right)(z_{k+1}), \tag{1.3}$$

$$\|\hat{u} + \hat{m}(z_k - z_{k+1})\|^2 \lesssim \hat{m}\left[\phi(z_k) - \phi(z_{k+1})\right], \tag{1.4}$$

$$\|\hat{u}\|^2 \lesssim \hat{m}^2 \|z_{k+1} - z_k\|^2. \tag{1.5}$$

(ii) If a key inequality fails during the execution of step (i), increase $\hat{m}$ and try step (i) again.

[a] The notation $\lesssim$ means that the inequality holds up to some multiplicative constants on the right-hand-side.

To find z_{k+1} in step 1, CF.APD specifically applies a curvature-free accelerated composite gradient (CF.ACG) algorithm to the subproblem $\min_{z\in\mathbb{R}^n}\{\phi(z) + \hat{m}\|z - z_k\|^2\}$ until a special set of inequalities holds. During the execution of CF.ACG, several key inequalities are also checked to ensure convergence and the execution is halted if at least one of these inequalities does not hold. These inequalities are always guaranteed to hold when $\hat{m} \geq m$ but may fail to hold when $\hat{m} < m$.

It is worth mentioning that the main difficulties preventing the extension of existing complexity-optimal methods to curvature-free ones is their dependence on *global* topological conditions that strongly depend on the knowledge of (m, M), e.g., (1.2), convexity of f, or knowledge of the Lipschitz modulus of ∇f. Hence, one of the novelties of CF.APD is its ability to relax these conditions to a finite set of *local* topological conditions that only depend on the generated sequence of iterates.

Contributions. Given a starting point $z_0 \in \operatorname{dom} h$ and a tolerance $\rho > 0$, it is shown that CF.APD, with an initial estimate of $\hat{m} = \rho$, obtains a pair $(\bar{z}, \bar{v}) \in \operatorname{dom} h \times \mathbb{R}^n$ satisfying the approximate stationarity condition

$$\bar{v} \in \nabla f(\bar{z}) + \partial h(\bar{z}), \quad \|\bar{v}\| \leq \bar{\rho} \tag{1.6}$$

in $\tilde{\mathcal{O}}(\sqrt{mM}[\phi(z_0) - \phi_*]/\rho^2)$ CF.ACG iterations/resolvent evaluations[1] when f is nonconvex and $\tilde{\mathcal{O}}(\sqrt{M}/\sqrt{\rho})$ CF.ACG iterations when f is convex. Both complexity bounds are optimal, up to logarithmic terms, and it appears to be the first time that a curvature-free method has obtained such bounds. Improved iteration complexity bounds are also obtained when ϕ_* or the distance from z_0 to the optimality set is known. It worth mentioning that all of these bounds are obtained under the mild assumption that the optimal value in (1.1) is finite and does not assume the boundedness of $\operatorname{dom} h$ (cf. [20, 32]) nor that an optimal solution of (1.1) even exists.

In addition to the development of CF.APD, some details are given regarding how CF.APD could be used in other existing optimization frameworks, including min-max smoothing and penalty frameworks for constrained optimization. The main advantages of these resulting frameworks is that (i) they are curvature-free and (ii) they have improved complexities in the convex regime without requiring any adjustments to their inputs.

[1] The notation $\tilde{\mathcal{O}}(\cdot)$ ignores any terms that logarithmically depend on the tolerance ρ.

Finally, numerical experiments are given to support the practical efficiency of CF.ADP on some randomly generated problem instances. These experiments specifically show that CF.APD vastly outperforms several existing curvature-free methods in practice.

Literature Review. We first discuss some accelerated methods for finding stationary points of (1.1) under the assumption that m and M are known. To keep our notation concise, we will make use of the scalars

$$(1.7) \qquad \Delta_0 := \phi(z_0) - \inf_{z\in\mathbb{R}^n} \phi(z), \quad d_0 := \inf_{z_*\in\mathbb{R}^n} \left\{ \|z_0 - z_*\| : \phi(z_*) = \inf_{z\in\mathbb{R}^n} \phi(z) \right\}.$$

One of the earliest accelerated gradient method for finding approximate stationary points of (1.1) is found in [8]. Under the assumption that $\operatorname{dom} h$ is bounded, [8] presented an accelerated method that obtains a point as in (1.6) in $\mathcal{O}(MmD_h^2 + [Md_0/\rho]^{2/3})$ iterations, where D_h denotes the diameter of h. Motivated by the developments in [8], other papers, e.g., [4,6,18,30], developed similar accelerated methods under different assumptions on f and h. Recently, [13] proposed an accelerated inexact proximal point method that has an optimal iteration complexity bound of $\mathcal{O}(\sqrt{Mm}\Delta_0/\rho^2)$ when f is weakly convex, but has no advantage when f is convex. The work in [14] proposed an adaptive version of this method in which (m, M) were estimated locally, but a lower bound for $\max\{m, M\}$ was still required. A version of [13] in which the outer proximal point scheme is replaced with an accelerated one was examined in [19].

We now discuss curvature-free methods for finding stationary points of (1.1), i.e., under the assumption that m and M are not known. One of the most well-known curvature-free algorithms for finding stationary points of (1.1) is the proximal gradient descent (PGD) method with backtracking line search. In [27], it was shown that this method obtains a $\mathcal{O}(\rho^{-2})$ complexity when f is weakly-convex and a $\mathcal{O}(\rho^{-1})$ bound when f is convex. More recently, [9] presented a curvature-free extension of the algorithm in [8] which can handle both Lipschitz continuous gradients of f and Hölder continuous ones. In a separate line of research, [20] presented a accelerated method whose main steps are based off of the FISTA algorithm in [3]. A variant of this method, with improved iteration complexity bounds in the convex setting, was examined in [32].

For convenience, we compare in Table 1.1 the detailed iteration complexity bounds of the curvature-free methods listed above with two instances of CF.APD. Specifically, PGD is the adaptive proximal gradient descent method in [27], UPF is the UPFAG method in [9], NCF is the ADAP-NC-FISTA method in [20], VRF is the VAR-FISTA method in [32], APD$\{\rho\}$ (resp. APD$\{\rho^2\}$) is CF.APD with initial estimate $\hat{m} = \rho$ (resp. $\hat{m} = \rho^2$).

Notice, in particular, that the analysis for UPFAG does include an iteration complexity bound for finding stationary points when f is convex, while NCF and VRF suffer from the requirement that $\operatorname{dom} h$ must be bounded. Moreover, up until to this point in time, PGD was the only curvature-free algorithm with an established iteration complexity bound for the unbounded case when f is convex. In the nonconvex setting, none of the curvature-free methods prior to this work were able to obtain the optimal complexity bound in [13].

Finally, it is worth mentioning some tangentially related works. The developments in [12, 16] strongly influenced and motivated the technical developments of

Algorithm	f convex	f nonconvex	$D_h < \infty$
PGD	$\mathcal{O}\left(\frac{M^{3/2}d_0}{\rho}\right)$	$\mathcal{O}\left(\frac{M^2\Delta_0}{\rho^2}\right)$	No
UPF	N/A	$\mathcal{O}\left(\frac{M\Delta_0}{\rho^2}\right)$	No
NCF	$\mathcal{O}\left(\frac{M^{2/3}[\Delta_0^{1/3}+d_0^{2/3}]}{\rho^{2/3}} + \frac{MD_h}{\rho}\right)$	$\mathcal{O}\left(mM^2\left[\frac{mD_h^2+\Delta_0}{\rho^2}\right]\right)$	Yes
VRF	$\mathcal{O}\left(\frac{M^{2/3}[\Delta_0^{1/3}+D_h^{2/3}]}{\rho^{2/3}}\right)$	$\mathcal{O}\left(mM^2D_h^2\left[\frac{1+m^2}{\rho^2}\right]\right)$	Yes
APD$\{\rho\}$	$\tilde{\mathcal{O}}\left(\sqrt{M}\left[\frac{1+\min\{\Delta_0,d_0^2\}}{\sqrt{\rho}}\right]\right)$	$\tilde{\mathcal{O}}\left(\frac{\sqrt{mM}\Delta_0}{\rho^2}\right)$	No
APD2$\{\rho^2\}$	$\tilde{\mathcal{O}}\left(\sqrt{M}\left[\frac{1+\rho^2\min\{\Delta_0,d_0^2\}}{\rho}\right]\right)$	$\tilde{\mathcal{O}}\left(\frac{\sqrt{mM}\Delta_0}{\rho^2}\right)$	No

TABLE 1.1

Comparison of iteration complexity bounds of various curvature-free accelerated composite optimization algorithms for finding ρ-stationary points as in (1.6). It is assumed that d_0, Δ_0, m, and M are not known but $M \geq m$. The scalar D_h denotes the diameter of $\operatorname{dom} h$.

both CF.ACG and CF.APD. Relative prox-stationarity criteria, such as (1.5), were previously analyzed in [31] and, more recently, in [1, 21, 23–26]. Papers [10, 11, 29] present curvature-free methods that obtain optimal complexity bounds for minimizing the objective function of (1.1) (rather than finding stationary points) when f is convex and $h \equiv 0$.

Summary of Contents. Section 2 gives some background material. Section 3 presents CF.ACG, CF.APD, and their iteration complexity bounds. Section 4 describes how CF.APD can be used in existing optimization frameworks. Section 5 presents some numerical experiments. Section 6 gives some concluding remarks. Two appendices follow after the above sections.

Notation. $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of nonnegative and positive real numbers, respectively. $\mathbb{R}^n$ denotes an n-dimensional Euclidean space with inner product and norm denoted by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$, respectively. $\operatorname{dist}(x, X)$ denotes the Euclidean distance of a point x to a set X. For any $t > 0$, we denote $\log_1^+(t) := \max\{\log t, 1\}$. For a function $h : \mathbb{R}^n \to (-\infty, \infty]$ we denote $\operatorname{dom} h := \{x \in \mathbb{R}^n : h(x) < +\infty\}$ to be the domain of h. Moreover, h is said to be proper if $\operatorname{dom} h \neq \emptyset$. The set of all lower semi-continuous proper convex functions defined in $\mathbb{R}^n$ is denoted by $\overline{\text{Conv}}\,(\mathbb{R}^n)$. The subdifferential of a proper function $h : \mathbb{R}^n \to (-\infty, \infty]$ is defined by

$$(1.8) \qquad \partial_\varepsilon h(z) := \{u \in \mathbb{R}^n : h(z') \geq h(z) + \langle u, z' - z\rangle - \varepsilon, \quad \forall z' \in \mathbb{R}^n\}$$

for every $z \in \mathbb{R}^n$. The classical subdifferential, denoted by $\partial h(\cdot)$, corresponds to $\partial_0 h(\cdot)$. If ψ is a real-valued function which is differentiable at $\bar{z} \in \mathbb{R}^n$, then its affine/linear approximation $\ell_\psi(\cdot, \bar{z})$ at $\bar{z}$ is given by

$$(1.9) \qquad \ell_\psi(z; \bar{z}) := \psi(\bar{z}) + \langle\nabla\psi(\bar{z}), z - \bar{z}\rangle \quad \forall z \in \mathbb{R}^n.$$

2. Background.

This section gives some necessary background for presenting CF.ACG and CF.APD. More specifically, Subsection 2.1 describes and comments on the problem of interest, while Subsection 2.2 presents a general proximal descent scheme which serves as a template for CF.APD.

2.1. Problem of Interest.

To reiterate, we are interested in the following composite optimization problem:

Problem $\mathcal{CO}$: Given $\rho \in \mathbb{R}_{++}$ and a function $\phi = f + h$ satisfying:
$\langle \texttt{A1} \rangle$ $h \in \overline{\text{Conv}}\,(\mathbb{R}^n)$,
$\langle \texttt{A2} \rangle$ f is continuously differentiable on an open set $\Omega \supseteq \operatorname{dom} f$, and ∇f is $\mathcal{M}$-Lipschitz continuous on $\operatorname{dom} h$ for some $\mathcal{M} \in \mathbb{R}_{++}$,
$\langle \texttt{A3} \rangle$ $\phi_* = \inf_{z \in \mathbb{R}^n} \phi(z) > -\infty$,
find a pair $(\bar{z}, \bar{v}) \in \operatorname{dom} h \times \mathbb{R}^n$ satisfying (1.6).

Of the three above assumptions, only $\langle \texttt{A1} \rangle$ is a necessary condition that is used to ensure CF.APD is well-defined. Assumptions $\langle \texttt{A2} \rangle$–$\langle \texttt{A3} \rangle$, on the other hand, are sufficient conditions that are used to show that CF.APD stops in a finite number of iterations. It is possible to replace assumption $\langle \texttt{A2} \rangle$ with more general smoothness conditions (e.g., Hölder continuity) at the cost of a possibly more complicated analysis. It is known[2] that assumption $\langle \texttt{A2} \rangle$ holds if and only if

$$|f(z) - \ell_f(z; z')| \le \frac{\mathcal{M}}{2} \|z - z'\|^2, \quad \forall z, z' \in \operatorname{dom} h, \tag{2.1}$$

which implies $(\mathcal{M}, \mathcal{M})$ is a curvature pair of ϕ.

We now give a few comments about criterion (1.6). First, it is related to the directional derivative of ϕ:

$$\begin{aligned}
\min_{\|d\|=1} \phi'(z; d) &= \min_{\|d\|=1} \max_{\zeta \in \partial h(z)} \langle \nabla f(z) + \zeta, d \rangle = \max_{\zeta \in \partial h(z)} \min_{\|d\|=1} \langle \nabla f(z) + \zeta, d \rangle \\
&= \max_{\zeta \in \partial h(z)} \min_{\|d\| \le 1} \langle \nabla f(z) + \zeta, d \rangle = \min_{\|d\| \le 1} \max_{\zeta \in \partial h(z)} \langle \nabla f(z) + \zeta, d \rangle \\
&= -\min_{\zeta \in \partial h(z)} \|\nabla f(z) + \zeta\| = -\text{dist}(0, \nabla f(z) + \partial h(z)).
\end{aligned}$$

Consequently, if $\bar{z} \in \operatorname{dom} h$ is local minimum of ϕ then $\min_{\|d\|=1} \phi'(\bar{z}; d) \ge 0$ and the above relation implies that (1.6) holds with $\rho = 0$. That is, (1.6) is a necessary condition for local optimality of a point $\bar{z} \in \operatorname{dom} h$. Second, when f is convex then (1.6) with $\rho = 0$ implies that $0 \in \nabla f(\bar{z}) + \partial h(\bar{z}) = \partial \phi(\bar{z})$ and $\bar{z}$ is a global minimum. In view of the first comment, (1.6) is equivalent to global optimality of a point $\bar{z} \in \operatorname{dom} h$ when f is convex.

2.2. General Proximal Descent Scheme.

Our interest in this subsection is the general proximal descent scheme in Algorithm 2.1, which follows the ideas in (1.3)–(1.5). Its iteration scheme will serve as a template for the CF.APD presented in Subsection 3.2.

Before presenting the properties of Algorithm 2.1, let us give a few comments about its steps. First, (2.2)–(2.4) are analogous to (1.3)–(1.5) in view of assumption $\langle \texttt{A1} \rangle$. Second, if $f + m_{k+1}\|\cdot\|^2$ is convex and $u_{k+1} = 0$ then (2.2) implies that

$$z_{k+1} = \operatorname*{argmin}_{z \in \mathbb{R}^n} \left\{ \phi(z) + m_{k+1} \|z - z_{k+1}\|^2 \right\},$$

which is a proximal point update with stepsize $1/(2m_{k+1})$. Third, (2.3) implies that Algorithm 2.1 is a descent scheme, i.e., $\phi(z_{k+1}) \le \phi(z_k)$ for $k \ge 0$. Hence, in view of the second comment, this justifies its qualifier as a "proximal descent" scheme.

[2] The proof of the forward direction is well-known (see, for example, [2, 28]) while the proof of the reverse direction can be found in [12, Proposition 2.1.55].

Algorithm 2.1 General Proximal Descent Scheme

```
Data: $(f,h)$ as in ⟨A1⟩–⟨A3⟩, $z_0 \in \operatorname{dom} h$;
Parameters: $\theta \in \mathbb{R}_+$;
for $k \leftarrow 0,1,\ldots$ do
    find $(z_{k+1},u_{k+1})$ and $m_{k+1} \in \mathbb{R}_{++}$ satisfying
    (2.2)    $u_{k+1} \in \nabla f(z_{k+1}) + 2m_{k+1}(z_{k+1}-z_k) + \partial h(z_{k+1})$,
    (2.3)    $\|u_{k+1} + 2m_{k+1}(z_k - z_{k+1})\|^2 \le 2\theta m_{k+1}\left[\phi(z_k)-\phi(z_{k+1})\right]$,
    (2.4)    $\|u_{k+1}\|^2 \le m_{k+1}^2\|z_{k+1}-z_k\|^2$.
```

It is also worth mentioning that (2.3)–(2.4) are variants of conditions in existing literature. More specifically, a version of (2.3) can be found in the descent scheme of [14], while (2.4) is can be found in the GIPP framework of [13] with $\sigma = 1$, $\tilde{\varepsilon} = 0$, and $v_{k+1} = u_{k+1}/m_{k+1}$. However, the addition of condition (2.2) appears to be new.

We now present the most important properties of Algorithm 2.1. The first result supports the importance of conditions (2.2)–(2.3).

Lemma 2.1. *Given $z_0 \in X$, let $\{(u_{k+1}, z_{k+1})\}_{k\ge 0}$ denote a sequence of iterates satisfying (2.2)–(2.3). Moreover, let Δ_0 be as in (1.7), and define*

$$v_{k+1} := u_{k+1} + 2m_{k+1}(z_k - z_{k+1}), \quad \Lambda_{k+1} := \sum_{j=0}^{k} \frac{1}{m_{j+1}}, \quad \forall k \ge 0.$$

Then, for every $k \ge 0$:

(a) $v_{k+1} \in \nabla f(z_{k+1}) + \partial h(z_{k+1})$*;*

(b) $\min\limits_{0\le j\le k} \|v_{j+1}\|^2 \le 2\theta\Delta_0\Lambda_{k+1}^{-1}$*.*

Proof. (a) This follows immediately from (2.2)and the definition of v_{k+1}.

(b) Using (2.3) at 0 to k, the definition of v_{k+1}, and the definition of ϕ_*, we have that

$$\begin{aligned}\Lambda_{k+1} \min_{0\le j\le k} \|v_{j+1}\|^2 &\le \sum_{j=0}^{k} \frac{\|v_{j+1}\|^2}{m_{j+1}} \overset{(2.3)}{\le} 2\theta \sum_{j=0}^{k} \left[\phi(z_j) - \phi(z_{j+1})\right] \\ &= 2\theta\left[\phi(z_0) - \phi(z_{k+1})\right] \le 2\theta\left[\phi(z_0) - \phi_*\right] = 2\theta\Delta_0. \qquad \square\end{aligned}$$

Notice that Lemma 2.1(b) implies that if $\lim_{k\to\infty} \Lambda_{k+1} \to \infty$ then we have that $\lim_{k\to\infty} \min_{j\le k} \|v_{j+1}\| \to 0$. Moreover, if $\sup_{k\ge 0} m_{k+1} < \infty$ then for any $\rho > 0$, there exists some finite $j \ge 0$ such that $\|v_{j+1}\| \le \rho$.

The next result shows that if m_{k+1} is bounded relative to the global topology of f, and conditions (2.2)–(2.4) hold, then a more refined bound of $\min_{j\le k}\|v_{j+1}\|$ can be obtained. To keep the notation concise, we make use of the following quantity:

$$(2.5) \qquad R_{\nu,m'}(z') := \inf_{z\in\mathbb{R}^n} \left\{ R_{\nu,m}(z,z') := \frac{\phi(z) - \phi_*}{\nu m'} + \frac{1}{2}\|z - z'\|^2 \right\}.$$

It is easy to see that $R_{\nu,m'}(z')$ is the Moreau envelope of $\phi/(\nu m')$ at z' shifted by $-\phi_*/(\nu m')$.

Lemma 2.2. *Given $z_0 \in X$, let $\{(v_{k+1}, z_{k+1}, \Lambda_{k+1})\}_{k\ge 0}$ be as in Lemma 2.1. Moreover, suppose (2.4) holds for every $k \ge 0$ and that there exists $\tilde{m} > 0$ such* *that $f + \tilde{m}\|\cdot\|^2/2$ is convex. If $\inf_{k\geq 0} m_{k+1} \geq \tilde{m}$ and $\sup_{k\geq 0} m_{k+1} \leq (1+\nu)\tilde{m}$ for some $\nu > 0$, then for $k \geq 0$ it holds that*

$$\phi(z_{k+1}) + \frac{m_{k+1}}{2}\|z_{k+1} - z_k\|^2 \leq \inf_{z\in\mathbb{R}^n}\left\{\phi(z) + \frac{\nu\tilde{m}}{2}\|z - z_k\|^2\right\}, \tag{2.6}$$

and for $k \geq 1$ it holds that

$$\min_{1\leq j\leq k}\|v_{j+1}\|^2 \leq 2\theta\nu\tilde{m}\left[\frac{R_{\nu,\tilde{m}}(z_0)}{\Lambda_{k+1} - m_1^{-1}}\right]. \tag{2.7}$$

Proof. Using the assumption that $m_{k+1} \geq \tilde{m}$ and (2.2), we have that $f(\cdot) + m_{k+1}\|\cdot - z_k\|^2$ is $\tilde{m}$-strongly convex and, hence,

$$\begin{aligned} u_{k+1} &\in \nabla f(z_{k+1}) + 2m_{k+1}(z_{k+1} - z_k) + \partial h(z_{k+1}) \\ &= \nabla f(z_{k+1}) - \tilde{m}(z_{k+1} - z_{k+1}) + 2m_{k+1}(z_{k+1} - z_k) + \partial h(z_{k+1}) \\ &= \partial\left(\phi - \frac{\tilde{m}}{2}\|\cdot - z_{k+1}\|^2 + m_{k+1}\|\cdot - z_k\|^2\right)(z_{k+1}). \end{aligned} \tag{2.8}$$

Using (2.8), (2.4), and the bound $\langle a, b\rangle \geq -\|a\|^2/(2m_{k+1}) - m_{k+1}\|b\|^2/2$ for any $a, b \in \mathbb{R}^n$, it holds for any $z \in \mathbb{R}^n$ that

$$\begin{aligned} &\phi(z) + m_{k+1}\|z - z_k\|^2 \\ &\overset{(2.8)}{\geq} \phi(z_{k+1}) + m_{k+1}\|z_k - z_{k+1}\|^2 + \frac{\tilde{m}}{2}\|z - z_{k+1}\|^2 + \langle u_{k+1}, z - z_{k+1}\rangle \\ &\geq \phi(z_{k+1}) + m_{k+1}\|z_k - z_{k+1}\|^2 - \frac{1}{2m_{k+1}}\|u_{k+1}\|^2 + \frac{\tilde{m} - m_{k+1}}{2}\|z - z_{k+1}\|^2 \\ &\overset{(2.4)}{\geq} \phi(z_{k+1}) + \frac{m_{k+1}}{2}\|z_k - z_{k+1}\|^2 + \frac{\tilde{m} - m_{k+1}}{2}\|z - z_{k+1}\|^2. \end{aligned}$$

Re-arranging terms and using the assumption $m_{k+1} \leq (1+\nu)\tilde{m}$, we then have that

$$\phi(z_{k+1}) + \frac{m_{k+1}}{2}\|z_k - z_{k+1}\|^2 \leq \phi(z) + \frac{\nu\tilde{m}}{2}\|z - z_k\|^2,$$

which implies (2.6) as $z \in \mathbb{R}^n$ was arbitrary. To show (2.7), we use (2.6) at $k = 1$, (2.3), and the definition of v_{k+1} to conclude that

$$\begin{aligned} R_{\nu,\tilde{m}}(z_0) = \inf_{z\in\mathbb{R}^n}\left\{\frac{\phi(z) - \phi_*}{\nu\tilde{m}} + \frac{1}{2}\|z - z_0\|^2\right\} &\geq \frac{\phi(z_1) - \phi_*}{\nu\tilde{m}} + \frac{m_1}{2\nu\tilde{m}}\|z_1 - z_0\|^2 \\ &\overset{(2.6)}{\geq} \frac{\phi(z_1) - \phi(z_{k+1})}{\nu\tilde{m}} = \frac{\sum_{j=1}^k [\phi(z_j) - \phi(z_{j+1})]}{\nu\tilde{m}} \overset{(2.3)}{\geq} \frac{1}{2\theta\nu\tilde{m}}\sum_{j=1}^k \frac{\|v_{j+1}\|^2}{m_{j+1}} \\ &\geq \frac{\sum_{j=1}^k m_{j+1}^{-1}}{2\theta\nu\tilde{m}}\left(\inf_{1\leq j\leq k}\|v_{j+1}\|^2\right) = \frac{\Lambda_{k+1} - m_1^{-1}}{2\theta\nu\tilde{m}}\inf_{1\leq j\leq k}\|v_{j+1}\|^2. \end{aligned}$$

□

Similar to the previous lemma, the above result also implies that if $\lim_{k\to\infty}\Lambda_{k+1} \to \infty$ then we have that $\lim_{k\to\infty}\min_{j\leq k}\|v_{j+1}\| \to 0$. However, it is more general in the sense that the rate of convergence depends on $R_{\nu,\tilde{m}}(z_0)$ instead of Δ_0, and the former can be bounded as

$$R_{\nu,\tilde{m}}(z_0) \leq \min\left\{R_{\nu,\tilde{m}}(z_0, z_0), R_{\nu,\tilde{m}}(z_*, z_0)\right\} \leq \min\left\{\frac{\Delta_0}{\nu\tilde{m}}, \frac{d_0^2}{2}\right\}, \tag{2.9}$$

where z_* is an optimal solution of (1.1) closest to z_0 and (Δ_0, d_0) are as in (1.7). This fact will be of particular important when we establish an iteration complexity bound for CF.APD in the convex setting.

3. Curvature-Free Algorithms.

This section presents CF.ACG, CF.APD, and their iteration complexity bounds. More specifically, Subsection 3.1 presents CF.ACG, while Subsection 3.2 presents CF.APD.

3.1. CF.ACG Algorithm.

To be as concise as possible, we present CF.ACG as an algorithm for finding approximate stationary points of the composite optimization problem

$$\min_{x\in\mathbb{R}^n} \{\psi(x) := \psi^s(x) + \psi^n(x)\} \tag{3.1}$$

where (ψ^s, ψ^n) satisfy ⟨A1⟩–⟨A2⟩ with $(f, h, \mathcal{M}) = (\psi^s, \psi^n, L_*)$ where L_* is the *smallest* scalar such that $\nabla\psi^s$ is L_*-Lipschitz continuous. Similar to (2.1), note that this implies that

$$|\psi^s(x) - \ell_{\psi^s}(x; x')| \le \frac{L_*}{2}\|x - x'\|^2 \quad \forall x, x' \in \operatorname{dom}\psi^n. \tag{3.2}$$

In Subsection 3.2, we specialize the main properties of CF.APD (see Proposition 3.5) to the context of attacking (3.1) with $\psi^s = f/(2\hat{m}) + \|\cdot - \hat{z}\|^2$ and $\psi^n = h/(2\hat{m})$ for some $\hat{m} > 0$ and $\hat{z} \in \operatorname{dom} h$.

To begin, we give a high-level overview of the method, followed by the precise details of the key steps at the $(j+1)^{\text{th}}$ iteration. Some elucidating comments will also be given throughout.

Broadly speaking, given $y_0 = x_0 \in \operatorname{dom}\psi^n$ and initial estimate $(\mu, L_0) \in \mathbb{R}^2_{++}$, the $(j+1)^{\text{th}}$ iteration of CF.ACG consists of the following steps:

Iteration $(j+1)$**:**

(i) Employ a line search to find a *good* estimate $L_{j+1} \ge L_j$ of L_*.
(ii) Apply an μ-ACG update to obtain the next iterate.
(iii) Check several key inequalities for local convexity, and stop the method early with a *failure* status if one of these inequalities does not hold.
(iv) Check for the successful termination of CF.ACG.

Step (ii) specifically consists of applying the μ-ACG update

$$(\mu, L) \overset{j}{\mapsto} \begin{cases} \xi_j \leftarrow 1 + \mu A_j \text{ and find } a_j \text{ satisfying } a_j^2 = \frac{\xi_j(a_j + A_j)}{L}, \\ A_{j+1} \leftarrow A_j + a_j, \\ \tilde{x}_j \leftarrow \frac{A_j}{A_{j+1}} y_j + \frac{a_j}{A_{j+1}} x_j, \\ y_{j+1} \leftarrow \operatorname{argmin}_{y\in\mathbb{R}^n} \left\{ \ell_{\psi_s}(y; \tilde{x}_j) + \psi_n(y) + \frac{L+\mu}{2}\|y - \tilde{x}_j\|^2 \right\}, \\ x_{j+1} \leftarrow x_j + \frac{a_j}{1 + A_{j+1}\mu}\left[L(y_{j+1} - \tilde{x}_j) + \mu(y_{j+1} - x_j)\right]. \end{cases} \tag{3.3}$$

and computing the auxiliary quantities

$$(3.4)\quad (\mu, L) \overset{j}{\mapsto} \begin{cases} \tilde{u}_{j+1} \leftarrow \nabla\psi^s(y_{j+1}) - \nabla\psi^s(\tilde{x}_j) + (L+\mu)(\tilde{x}_j - y_{j+1}) \\ \tilde{q}_{j+1}(\cdot) \leftarrow \ell_{\psi^s}(\cdot, \tilde{x}_j) + \psi^n(\cdot) + \frac{\mu}{2}\|\cdot - \tilde{x}_j\|^2, \\ q^L_{j+1}(\cdot) \leftarrow \tilde{q}_{j+1}(y_{j+1}) + L\left\langle \tilde{x}_j - y_{j+1}, \cdot - y_{j+1}\right\rangle + \frac{\mu}{2}\|\cdot - y_{j+1}\|^2, \\ Q^L_{j+1}(\cdot) \leftarrow \frac{A_j}{A_{j+1}} Q^L_j(\cdot) + \frac{a_j}{A_{j+1}} q^L_j(\cdot). \end{cases}$$

for a given estimate $L = L_{j+1}$ of L_*, a positive scalar μ, and $A_0 = 0$. It is straightforward to show that the updates in (3.3) follow similar ones in other strongly-convex accelerated solvers such as [7, 23, 27, 28]. In (3.4), the residual $\tilde{u}_{j+1}$ serves as an approximation of $\operatorname{dist}(0, \nabla\psi^s(y_{j+1}) + \partial\psi^n(y_{j+1}))$, while the functions $\tilde{q}_{j+1}$, q^L_{j+1}, and Q^L_{j+1} will be necessary in step (i), step (iii), and the analysis of CF.ACG. It also is worth mentioning that these functions also appear in the motivating works [12, 16].

Moving on, for a given $\beta > 1$, step (i) specifically consists of finding the smallest integer $s \geq 0$ such that the inequalities

$$(3.5)\qquad (\mu, L) \overset{j}{\mapsto} \begin{cases} \psi_s(y_{j+1}) - \ell_{\psi_s}(y_{j+1}; \tilde{x}_k) \leq \frac{L}{2}\|y_{j+1} - \tilde{x}_j\|^2, \\ \frac{\mu A_{j+1}}{2}\|y_{j+1} - \tilde{x}_j\|^2 + \frac{\xi_{j+1}}{2}\|y_j - x_{j+1}\|^2 \\ \qquad \leq A_{j+1}\left[q^L_{j+1}(y_j) - \psi(y_{j+1})\right] + \frac{\xi_j}{2}\|y_j - x_j\|^2. \end{cases}$$

hold with $L = L_{j+1} = \beta^s L_j$, $(\xi_j, \tilde{x}_j, y_{j+1}, A_{j+1})$ computed as in (3.3), and q^L_{j+1} computed as in (3.4). This can easily be done by repeatedly checking conditions (3.5) on the sequence of ordered candidates $\{L_j \beta^s\}_{s=0}^{\infty}$. Also, notice that the first condition clearly holds for $L \geq L_*$ in view of (3.2). In later analyses, we show that the second condition, together with some local convexity conditions (specifically (3.6) below), will be enough to guarantee that our desired termination condition (specifically (3.7) below) will hold at some finite iteration $j \geq 0$. For now, we present the following important result about this condition.

Lemma 3.1. *Given $(\mu, L) \in \mathbb{R}^2_{++}$ and $(y_j, x_j, A_j) \in \operatorname{dom}\psi^n \times \mathbb{R}^n \times \mathbb{R}_{++}$, let the quadruple $(A_{j+1}, y_{j+1}, \tilde{x}_j, q^L_{j+1})$ be generated by (3.3) and (3.4). If $L \geq L_*$, then (3.5) holds.*

Its proof can be found in Appendix A, and it essentially shows that step (i) is well-defined as long as L_* is finite.

Next, step (iii) specifically consists of checking if the local convexity conditions

$$(3.6)\qquad (\mu, L) \overset{j}{\mapsto} \begin{cases} q^L_{j+1}(y_j) \leq \psi(y_j), \\ Q^L_{j+1}(y) \leq \psi(y) \quad \forall y \in \{y_j, y_{j+1}\}, \\ \psi(y_0) \geq \psi(y_{j+1}) + \left\langle \tilde{u}_{j+1}, y_0 - y_{j+1}\right\rangle, \end{cases}$$

hold. If (3.6) does not hold, then CF.ACG stops early and signals that it has *failed.* The next result gives a sufficient condition for (3.5) to hold.

Lemma 3.2. *Given $(\mu, L) \in \mathbb{R}^2_{++}$ and $(y_{j+1}, \tilde{x}_j) \in \operatorname{dom}\psi^n \times \mathbb{R}^n$, let the triple $(\tilde{q}_{j+1}, q^L_{j+1}, Q^L_{j+1})$ be generated by (3.6). If ψ^s is μ-strongly convex, then (3.6) holds.*

Proof. The proof of the first two conditions in (3.6) can be found in [12, Lemma B.0.1] and [12, Lemma B.0.3]. The third condition follows from the optimality condition of y_{j+1} in (3.3), the definition of $\tilde{u}_{j+1}$ in (3.3), and the fact that if ψ^s is convex then

$$\begin{aligned} \tilde{u}_{j+1} &= \nabla\psi^s(y_{j+1}) - \nabla\psi^s(\tilde{x}_j) + (L+\mu)(\tilde{x}_j - y_{j+1}) \\ &\in \nabla\psi^s(y_{j+1}) - \nabla\psi^s(\tilde{x}_j) + \nabla\psi(\tilde{x}_j) + \partial\psi^n(y_{j+1}) = \partial\psi(y_{j+1}). \end{aligned} \qquad \square$$

Finally, for a given $\theta > 2$ and $\sigma > 0$, step specifically (iv) consists of checking the termination conditions

$$(3.7) \qquad \begin{cases} \|\tilde{u}_{j+1} + y_0 - y_{j+1}\|^2 \le \theta \left[\psi(y_0) - \psi(y_{j+1}) + \frac{1}{2}\|y_{j+1} - y_0\|^2\right] \\ \|\tilde{u}_{j+1}\|^2 \le \sigma^2 \|y_{j+1} - y_0\|^2. \end{cases}$$

If (3.7) does hold, CF.ACG stops early and signals that it has *succeeded*. The significance of (3.7) comes from the fact that if CF.APD follows the iteration scheme of Algorithm 2.1, then (3.7) is directly analogous to (2.3)–(2.4) when $\psi(\cdot) = \phi(\cdot)/(2m_{k+1}) + \|\cdot - z_k\|^2/2$.

For the ease of future reference and discussion, pseudocode for CF.ACG is given in Algorithm 3.1. Note that the output variable $\mathcal{S}$ in Algorithm 3.1 represents the termination state of CF.ACG, where $\mathcal{S} = 0$ represents *failure* and $\mathcal{S} = 1$ represents *success*.

Algorithm 3.1 Curvature-Free Accelerated Composite Gradient (CF.ACG) Algorithm

```
Data: $(\psi^s, \psi^n)$ as in ⟨A1⟩–⟨A2⟩ with $(f, h) = (\psi^s, \psi^n)$, $y_0 \in \mathrm{dom}\,\psi^n$, $\mu \in \mathbb{R}_{++}$, $L_0 \in [\mu, \infty)$;
Parameters: $\sigma \in \mathbb{R}_{++}$, $\theta \in (2, \infty)$, $\beta \in (1, \infty)$;
Outputs: $(\bar{y}, \bar{u}, L, \mathcal{S}) \in \mathrm{dom}\,\psi^n \times \mathbb{R}^n \times \mathbb{R}_{++} \times \{0, 1\}$;
(x_0, A_0) ← (y_0, 0)
for $j \leftarrow 0, 1, \ldots$ do
    find the smallest integer $s \ge 0$ where (3.5) holds with $L = \beta^s L_j$
    $L_{j+1} \leftarrow \beta^s L_j$
    compute $(y_{j+1}, \tilde{u}_{j+1})$ as in (3.3) and (3.4) with $L = L_{j+1}$
    if (3.6) does not hold with $L = L_{j+1}$ then
        return $(y_{j+1},\ \tilde{u}_{j+1},\ L_{j+1}, 0)$
    if (3.7) does hold then
        return $(y_{j+1},\ \tilde{u}_{j+1},\ L_{j+1}, 1)$
```

The next result presents four important technical properties of CF.ACG.

Lemma 3.3. *Let $\{(y_{j+1}, \tilde{u}_{j+1})\}_{j \ge 0}$ and $\{(L_{j+1}, \xi_{j+1}, A_{j+1})\}_{j \ge 0}$ be generated by Algorithm 3.1 and let $\bar{L} := \max\{\beta L_*, L_0\}$. Then, for every $j \ge 0$:*

(a) $L_j \le L_{j+1} \le \bar{L}$;

(b) $A_{j+2} \ge (1/L_1) \prod_{i=1}^{j+1} [1 + \sqrt{\mu/(2L_i)}]$;

(c) $\tilde{u}_{j+1} \in \nabla\psi^s(y_{j+1}) + \partial\psi^n(y_{j+1})$;

(d) if $\xi_{j+1} \ge 4$, then $\|\tilde{u}_{j+1}\| \le 12\bar{L}\|y_{j+1} - y_0\|/\sqrt{\mu A_{j+1}}$.

Proof. (a) The fact that $L_j \le L_{j+1}$ is immediate. If $L_0 \ge L_*$, then Lemma 3.1 implies that $L = L_0$ always suffices to ensure (3.5) holds and, hence, $L_{j+1} = L_0$ for every $j \ge 0$. On the other hand, if $L_0 < L_*$, then then Lemma 3.1 implies that the most that L_{j+1} can be increased to is βL_*. Combining both cases yields the desired conclusion.

(b) See [12, Lemma B.0.2].

(c) The optimality of y_{j+1} implies that $(L_{j+1} + \mu)(\tilde{x}_j - y_{j+1}) \in \nabla\psi^s(\tilde{x}_j) + \partial\psi^n(y_{j+1})$, which implies the desired inclusion in view of the definition of $\tilde{u}_{j+1}$.

(d) See Appendix B. □

The first three properties above are well-known, while the last one appears to be new. The next result shows that if A_{j+1} is sufficiently large, then (3.7) holds and Algorithm 3.1 terminates successfully.

LEMMA 3.4. *Let $\bar{L}$ and $\{A_{j+1}\}_{j\geq 0}$ be as in Lemma 3.3. If, for some iteration $j \geq 1$,*

$$A_{j+1} \geq \frac{4\bar{L}}{\mu}\left[\frac{1}{\mu} + 36\bar{L}\min\left\{\frac{1}{\sigma^2}, \frac{4\theta}{\theta-2}\right\}\right] =: \mathcal{A}_{\mu,\bar{L}}(\sigma,\theta) \tag{3.8}$$

then (3.7) holds.

Proof. Using the fact that $A_{j+1} \geq 4\bar{L}/\mu^2$ and the fact that $\bar{L} \geq \mu$, we first have that $\xi_{j+1} \geq 1 + 4\bar{L}/\mu \geq 4$. Combining this bound, Lemma 3.3(d), and the fact that $A_{j+1} \geq 144\bar{L}^2/(\sigma^2\mu)$, we then have that

$$\|u_{j+1}\| \overset{\text{Lemma 3.3(d)}}{\leq} \frac{12\bar{L}}{\sqrt{\mu A_{j+1}}}\|y_{j+1} - y_0\| \leq \sigma\|y_{j+1} - y_0\|$$

and, hence, the second condition in (3.7) holds. To show the other inequality, we first let $\gamma := \sqrt{(\theta-2)/\theta}$. Using the fact that $A_{j+1} \geq 576\bar{L}^2/(\mu\gamma^2)$, Lemma 3.3(d), the fact that $\gamma \in (0,1)$, and the bound $\|a+b\|^2 \leq (1+\gamma)\|a\|^2 + (1+\gamma^{-1})\|b\|^2$ for $a, b \in \mathbb{R}^n$, we then have that

$$\begin{aligned}
\|u_{j+1}\|^2 &\overset{\text{Lemma 3.3(d)}}{\leq} \frac{144\bar{L}^2}{\mu A_{j+1}}\|y_{j+1} - y_0\|^2 \\
&\leq \frac{\gamma^2}{4}\|y_{j+1} - y_0\|^2 \overset{\gamma\in(0,1)}{\leq} \left(\frac{\gamma}{1+\gamma}\right)^2 \|y_{j+1} - y_0\|^2 \\
&\leq \left(\frac{\gamma}{1+\gamma}\right)^2 (1+\gamma)\|\tilde{u}_{j+1} + y_{j+1} - y_0\|^2 + \left(\frac{\gamma}{1+\gamma}\right)^2\left(1 + \frac{1}{\gamma}\right)\|u_{j+1}\|^2 \\
&= \frac{\gamma^2}{1+\gamma}\|\tilde{u}_{j+1} + y_{j+1} - y_0\|^2 + \frac{\gamma}{1+\gamma}\|u_{j+1}\|^2,
\end{aligned}$$

which implies $\|\tilde{u}_{j+1}\|^2 \leq \gamma^2\|\tilde{u}_{j+1} + y_{j+1} - y_0\|^2$. Using this bound, the definition of γ, and the third condition of (3.6), we have that

$$\begin{aligned}
2\left[\psi(y_0) - \psi(y_{j+1})\right] &\overset{(3.6)}{\geq} 2\left\langle u_{j+1}, y_0 - y_{j+1}\right\rangle \\
&= \|\tilde{u}_{j+1} + y_0 - y_{j+1}\|^2 - \|\tilde{u}_{j+1}\|^2 - \|y_0 - y_{j+1}\|^2 \\
&\geq (1-\gamma^2)\|\tilde{u}_{j+1} + y_0 - y_{j+1}\|^2 - \|y_0 - y_{j+1}\|^2 \\
&= \frac{2}{\theta}\|\tilde{u}_{j+1} + y_0 - y_{j+1}\|^2 - \|y_0 - y_{j+1}\|^2,
\end{aligned}$$

and, hence, the first condition in (3.7) holds. □

We are now ready give an iteration complexity bound for CF.ACG and a condition for guaranteeing its successful termination.

PROPOSITION 3.5. *The following properties hold about Algorithm 3.1:*

(a) it stops in

$$\left\lceil 1 + 2\sqrt{\frac{2\bar{L}}{\mu}}\log_1^+\left\{\bar{L}\mathcal{A}_{\mu,\bar{L}}(\sigma,\theta)\right\}\right\rceil; \tag{3.9}$$

iterations, where $\bar{L}$ and $\mathcal{A}_{\mu,\bar{L}}$ are as in Lemma 3.3 and (3.8), respectively.

(b) if ψ^s is μ-strongly convex, then it always terminates in line 9 with a quadruple $(y_{j+1}, \tilde{u}_{j+1}, L_{j+1}, \mathcal{S})$ satisfying (3.7) and

$$\tilde{u}_{j+1} \in \nabla\psi^s(y_{j+1}) + \partial\psi^n(y_{j+1}), \quad \overline{L} \geq L_{j+1} \geq L_0, \quad \mathcal{S} = 1.$$

Proof. (a) Let J be the quantity in (3.9) minus one and $\mathcal{A} = \mathcal{A}_{\mu,\bar{L}}(\sigma, \theta)$ be as in (3.8). Now, suppose Algorithm 3.1 has not terminated by iteration $J + 1$. It follows from Lemma 3.3(a)–(b) that

$$A_{J+1} \geq \frac{1}{L_1} \prod_{i=1}^{J} \left(1 + \sqrt{\frac{\mu}{2L_i}}\right) \geq \frac{1}{\bar{L}} \left(1 + \sqrt{\frac{\mu}{2\bar{L}}}\right)^J. \tag{3.10}$$

Using the fact that $J \geq 2\sqrt{2\bar{L}/\mu}\log(\bar{L}\mathcal{A})$ from the definition in (3.9), (3.10), the bound $\mu \leq \bar{L}$, and the fact that $\log(1 + t) \geq t/2$ on $t \in [0, 1]$, it holds that

$$\log(\bar{L}\mathcal{A}) \leq \frac{J}{2}\sqrt{\frac{\mu}{2\bar{L}}} \leq J \log\left(1 + \sqrt{\frac{\mu}{2\bar{L}}}\right) \overset{(3.10)}{\leq} \log(\bar{L}A_{J+1})$$

which implies $A_{J+1} \geq \mathcal{A}$. Hence, it follows from Lemma 3.4 that (3.7) holds. In view of line 9 of Algorithm 3.1 this implies that termination has to have occurred at or before iteration $J + 1$, which contradicts our initial assumption. Thus, Algorithm 3.1 must have terminated by iteration $J + 1$.

(b) This follows immediately from line 9, Lemma 3.2, Lemma 3.3(a) and (c), and the fact that the algorithm stops in a finite number of iterations from part (a). ∎

Before ending this subsection, we give three closing comments about Algorithm 3.1. First, note that $\tilde{q}_{j+1}$, q^L_{j+1}, and Q^L_{j+1} are quadratic in their arguments, have a Hessian of μI, and, hence, are of the form $a + \langle b, \cdot\rangle + \mu\|\cdot\|^2/2$ for some $a \in \mathbb{R}$ and $b \in \mathbb{R}^n$. Thus, storage of each function can done by using $\Theta(n)$ space to store the relevant quantities $\{a, b\}$. Second, the update for y_{j+1} is equivalent to

$$y_{j+1} = \operatorname*{argmin}_{y \in \mathbb{R}^n} \left\{ \frac{\psi^n(y)}{L + \mu} + \frac{1}{2}\left\| y - \left(\tilde{x}_j - \frac{\nabla\psi_s(\tilde{x}_j)}{L + \mu}\right)\right\|^2 \right\}$$

and, hence, its computation only requires a prox-oracle for the function $\lambda\psi^n$ for any $\lambda > 0$. Third, the total number of additional ACG steps, i.e., (3.3), needed to find s in the ordered sequence of candidates $\{L_j\beta^s\}_{s=0}^{\infty}$ over all calls of line 3 is at most $1 + \log_1^+(\beta L_*/L_0)$. Since this number is on the same order of magnitude as in (3.9), we do not count these line search iterations as they do materially influence the complexity bound for CF.APD in the next subsection. Finally, it follows from the proof of Proposition 3.5 that the larger $\sum_{i=0}^{j} A_{j+1}$ is the faster CF.APD terminates. Hence, it is of general interest that L_0 be made small in practice.

3.2. CF.APD Algorithm. We are now ready to present the CF.APD and its iteration complexity bound. Its outer iteration scheme is based around the steps in Algorithm 2.1.

Broadly speaking, CF.APD is a double-loop method consisting of *outer iterations* and (possibly) several *inner iterations* per outer iteration. Given $z_0 \in \operatorname{dom} h$ and an estimate $(m_0, M_0) \in \mathbb{R}^2_{++}$, the $(k + 1)^{\text{th}}$ outer iteration of CF.APD broadly follows the approach in Algorithm 2.1 and consists of the following steps:

Iteration $(k+1)$**:**

(i) Repeatedly call CF.ACG on the prox-subproblem

$$\min_{z\in\mathbb{R}^n}\left\{\phi(z)+\hat{m}\|\cdot-z_k\|^2\right\}$$

for nondecreasing values of $\hat{m}\geq m_k$, until it stops successfully at some $\hat{m}=m_{k+1}$ with output $(z_{k+1},\tilde{u}_{k+1},L_{k+1},1)$.

(ii) Check for the successful termination of CF.APD.

(iii) If CF.APD has not terminated, update the estimate of M as $M_{k+1}\leftarrow 2m_{k+1}(L_{k+1}-1)$.

The inner iterations, on the other hand, refer to the iterations performed by CF.ACG. Given a free parameter $\theta\in(2,\infty)$ and an ACG line search parameter $\beta\in(1,\infty)$, the call in step (i) to Algorithm 3.1 is specifically of the form

$$(m,M)\overset{k}{\mapsto}\begin{cases}[\psi^s(\cdot),\psi^n(\cdot)]\leftarrow\left[\frac{f(\cdot)}{2m}+\frac{1}{2}\|\cdot-z_k\|^2,\,\frac{h(\cdot)}{2m}\right],\\ \textbf{call}\text{ Algorithm 3.1 with data }(\psi^s,\psi^n),\;z_k,\;(\frac{1}{2},\frac{M}{2m}+1)\\ \quad\text{and parameters }1/4,\;\theta,\;\beta\text{ to obtain an output tuple}\\ \quad(z_{k+1},\tilde{u}_{k+1},L_{k+1},\mathcal{S}_{k+1})\in\operatorname{dom}h\times\mathbb{R}^n\times\mathbb{R}_{++}\times\{0,1\},\end{cases}\tag{3.11}$$

for some curvature pair estimate (m,M). Moreover, the termination of the method in step (ii) specifically occurs when $\|2m_{k+1}(\tilde{u}_{k+1}+z_k-z_{k+1})\|\leq\rho$, which we later show is sufficient to solve Problem $\boldsymbol{\mathcal{CO}}$.

For the ease of future reference and discussion, the pseudocode for CF.APD is given in Algorithm 3.2.

Algorithm 3.2 Curvature-Free Accelerated Proximal Descent (CF.APP) Algorithm

```
Data: $(f,h)$ as in ⟨A1⟩–⟨A3⟩, $z_0\in\operatorname{dom}h$, $m_0\in\mathbb{R}_{++}$, $M_0\in[m_0,\infty)$, $\rho\in\mathbb{R}_{++}$;
Parameters: $\theta\in(2,\infty)$, $\alpha\in(1,\infty)$, $\beta\in(1,\infty)$;
Outputs: $(\bar{z},\bar{v})\in\operatorname{dom}h\times\mathbb{R}^n$;
for $k\leftarrow 0,1,\ldots$ do
    find the smallest integer $s\geq 0$ such that the output obtained by (3.11) with
  $(m,M)=(m_k\alpha^s,M_k)$ is of the form $(z_{k+1},\tilde{u}_k,L_{k+1},1)$
    $m_{k+1}\leftarrow m_k\alpha^s$
    $v_{k+1}\leftarrow 2m_{k+1}(\tilde{u}_{k+1}+z_k-z_{k+1})$
    if $\|v_{k+1}\|\leq\rho$ then
        return $(z_{k+1},\,v_{k+1})$
    $M_{k+1}\leftarrow 2m_{k+1}(L_{k+1}-1)$
```

We now present three important properties about Algorithm 3.2. To ensure that the resulting properties account for the possible asymmetry in (1.2), we make use of the scalars

$$m_*:=\operatorname*{argmin}_{z,z'\in\operatorname{dom}h,\;t\geq 0}\left\{t: f(z)-\ell_f(z;z')\geq-\frac{t}{2}\|z-z\|^2\right\},$$

$$M_*:=\operatorname*{argmin}_{z,z'\in\operatorname{dom}h,\;t\geq 0}\left\{t: f(z)-\ell_f(z;z')\leq\frac{t}{2}\|z-z\|^2\right\},$$

which are the smallest possible first and second components of a curvature pair of f.

Lemma 3.6. *Define the scalars*

$$\begin{gathered}\overline{m} := \max\{m_0, (\alpha+\beta)m_*\}, \quad \overline{M} := \max\{M_0, (\alpha+\beta)M_*\}, \\ P_0 := \frac{5\beta(\overline{M}+\overline{m})}{m_0}, \quad C_0 := \sqrt{m_0 P_0}\log_+^1\left\{P_0\mathcal{A}_{\frac{1}{2},P_0}\left(\frac{1}{4},\theta\right)\right\},\end{gathered} \tag{3.12}$$

and let $u_{k+1} := 2m_{k+1}\tilde{u}_{k+1}$ for $k \geq 0$. Then, for every $k \geq 0$, the following statements hold about Algorithm 3.2 and its iterates:

(a) $m_k \leq m_{k+1} \leq \overline{m}$ and $M_k \leq M_{k+1} \leq 3\beta(\overline{M}+\overline{m})$;

(b) (u_{k+1}, z_{k+1}) satisfies (2.2)–(2.4) and, hence, Algorithm 3.2 is an instance of Algorithm 3.1;

(c) the k-th outer iteration of Algorithm 3.2 performs at most $\lceil 4C_0 m_{k+1}^{-1/2}\rceil$ inner iterations.

Proof. (a) We first note that the k^{th} successful call of CF.ACG is such that ψ^s in (3.11) has the curvature pair

$$\left(\max\left\{0, \frac{m_*}{2m_{k+1}} - 1\right\}, \frac{M_*}{2m_{k+1}} + 1\right). \tag{3.13}$$

Hence, it follows from line 2 of Algorithm 3.2, Lemma 3.2, the curvature pair of ψ^s in (3.13), and the definition of $\overline{m}$ imply that (i) CF.ACG is called with m_{k+1} being at most $\overline{m}$ in view of the choice of $\mu = 1/2$ and (ii) $m_k \leq m_{k+1}$.

To show the bound on M_k, note that the curvature pair of ψ^s in (3.13) implies that $\nabla\psi^s$ is L_*-Lipschitz continuous where $L_* = (M_* + m_*)/(2m_{k+1}) + 1$. It then follows from the previous bound on m_{k+1} and Proposition 3.5(b) with $\overline{L} \leq (\overline{M} + \overline{m})/(2m_{k+1}) + \beta$ that

$$\frac{M_k}{2m_{k+1}} + 1 \leq \frac{M_{k+1}}{2m_{k+1}} + 1 \leq \frac{\overline{M}+\overline{m}}{2m_{k+1}} + \beta \leq \frac{3\beta(\overline{M}+\overline{m})}{2m_{k+1}}$$

which immediately implies $M_{k+1} \geq M_k$ and $M_{k+1} \leq 3\beta(\overline{M}+\overline{m})$.

(b) Using Lemma 3.3(c) and (3.11), it holds that

$$\tilde{u}_{k+1} \in \frac{\nabla f(z_{k+1}) + \partial h(z_{k+1})}{2m_{k+1}} + (z_{k+1} - z_k)$$

which implies that $u_{k+1} \in \nabla f(z_{k+1}) + \partial h(z_{k+1})$ and, hence, that (2.2) holds. Now, since the fourth argument in the output of Algorithm 3.1 is 1, it follows that (3.7) holds with $(y_{k+1}, y_0) = (z_{k+1}, z_k)$, $\sigma = 1/4$, and $\psi(\cdot) = \phi(\cdot)/(2m_{k+1}) + \|\cdot - z_k\|^2/2$. In particular, the first condition of (3.7) implies

$$\|u_{k+1}\|^2 = 4m_{k+1}^2\|\tilde{u}_{k+1}\|^2 \overset{(3.7)}{\leq} m_{k+1}^2\|y_{j+1} - y_0\|^2,$$

which is exactly (2.3). The second condition of (3.7), on the other hand, implies that

$$\begin{aligned}\|u_{k+1} + 2m_{k+1}(z_k - z_{k+1})\|^2 &= 4m_{k+1}^2\|\tilde{u}_{k+1} + z_k - z_{k+1}\|^2 \\ &\overset{(3.7)}{\leq} 4\theta m_{k+1}^2\left[\psi(z_k) - \psi(z_{k+1}) + \frac{1}{2}\|z_{k+1} - z_k\|^2\right] \\ &= 2\theta m_{k+1}\left[\phi(z_k) - \phi(z_{k+1})\right],\end{aligned}$$

which is exactly (2.4). Combing both inequalities yields the desired conclusion.

(c) Using Proposition 3.5 and (3.11), it holds that the call to Algorithm 3.1 at iteration k (in (3.11) with $(m, M) = (m_{k+1}, M_{k+1})$) takes at most

$$\mathcal{I}_k := \left\lceil 1 + 4\sqrt{L_{k+1}} \log_+^1 \left\{ L_{k+1} \mathcal{A}_{\frac{1}{2}, L_{k+1}} \left(\frac{1}{4}, \theta \right) \right\} \right\rceil$$

inner iterations. Now, using part (a) and the fact that $m_0 \le m_{k+1}$, we have that

$$\begin{aligned} L_{k+1} = \frac{M_{k+1} + 2m_{k+1}}{2m_{k+1}} &\overset{(a)}{\le} \frac{3\beta(\overline{M} + \overline{m}) + 2\overline{m}}{2m_{k+1}} \\ &\le \frac{5\beta(\overline{M} + \overline{m})}{m_{k+1}} = \frac{m_0 P_0}{m_{k+1}}. \end{aligned} \tag{3.14}$$

Consequently, using the above bound, the fact that $m_0 P_0 / m_{k+1} \le P_0$, and the definition of C_0, we conclude that

$$\begin{aligned} \mathcal{I}_k &\le 4\sqrt{L_{k+1}} \log_+^1 \left\{ L_{k+1} \mathcal{A}_{\frac{1}{2}, L_{k+1}} \left(\frac{1}{4}, \theta \right) \right\} \\ &\overset{(3.14)}{\le} 4\sqrt{\frac{m_0 P_0}{m_{k+1}}} \log_+^1 \left\{ \frac{m_0 P_0}{m_{k+1}} \mathcal{A}_{\frac{1}{2}, \frac{m_0 P_0}{m_{k+1}}} \left(\frac{1}{4}, \theta \right) \right\} \\ &\le 4\sqrt{\frac{m_0 P_0}{m_{k+1}}} \log_+^1 \left\{ P_0 \mathcal{A}_{\frac{1}{2}, P_0} \left(\frac{1}{4}, \theta \right) \right\} = \frac{4C_0}{\sqrt{m_{k+1}}} \le \left\lceil \frac{4C_0}{\sqrt{m_{k+1}}} \right\rceil. \end{aligned}$$ □

We are now ready to give some iteration complexity bounds on CF.APD.

THEOREM 3.7. *Algorithm 3.2 stops and outputs a pair* $(\bar{z}, \bar{v}) = (z_{k+1}, v_{k+1})$ *solving Problem* $\boldsymbol{\mathcal{CO}}$ *in a finite number of inner iterations* $\overline{T}$*, where*

$$\overline{T} \le \left\lceil 8C_0 \sqrt{\left(1 + \frac{2\theta\overline{m}\Delta_0}{\rho^2}\right)\left(\frac{1}{m_0} + \frac{2\theta\Delta_0}{\rho^2}\right)} \right\rceil, \tag{3.15}$$

and the quantities $(\overline{m}, C_0)$ *and* Δ_0^* *are as in* (3.12) *and* (1.7)*, respectively. Moreover, if* $m_0 \ge m_*$ *then*

$$\overline{T} \le \left\lceil \frac{8C_0}{\sqrt{m_0}} \left(1 + \frac{\theta m_0^2 \min\left\{2\Delta_0, d_0^2\right\}}{\rho^2} \right) \right\rceil \tag{3.16}$$

where d_0 *is as in* (1.7).

Proof. The fact that Algorithm 3.2 stops with a solution to Problem $\boldsymbol{\mathcal{CO}}$ follows immediately from Lemma 3.6(b), Lemma 2.1, and the termination condition in line 5 of Algorithm 3.2. To show (3.15), we first notice that Lemma 2.1(b) and Lemma 3.3(a) imply that $\overline{T}$ is bounded above by a unique integer $K \ge 1$ satisfying

$$\sum_{k=0}^{K-1} \frac{1}{m_{k+1}} \ge \frac{2\theta\Delta_0}{\rho^2} > \sum_{k=0}^{K-2} \frac{1}{m_{k+1}} \ge \frac{K-1}{\overline{m}}. \tag{3.17}$$

Combining the above bound, Lemma 3.6(c), and the fact that $\|z\|_1 \le \sqrt{n}\|z\|_2$ for any

$z \in \mathbb{R}^n$, it follows that $\overline{T}$ is bounded as

$$\overline{T} \le \sum_{k=0}^{K-1}\left\lceil \frac{4C_0}{\sqrt{m_{k+1}}}\right\rceil \le 8C_0 \sum_{k=0}^{K-1}\frac{1}{\sqrt{m_{k+1}}} \le 8C_0\sqrt{K\sum_{k=0}^{K-1}\frac{1}{m_{k+1}}}$$

$$\le 8C_0\sqrt{K\left(\frac{1}{m_0}+\sum_{k=0}^{K-2}\frac{1}{m_{k+1}}\right)} \overset{(3.17)}{\le} 8C_0\sqrt{\left(1+\frac{2\theta\Delta_0\overline{m}}{\rho^2}\right)\left(\frac{1}{m_0}+\frac{2\theta\Delta_0}{\rho^2}\right)},$$

which is exactly (3.15). To show (3.16), suppose $m_0 \ge m_*$. It follows from the definition of m_* that all calls to Algorithm 3.1 return an output quadruple $(z, \tilde{u}, L, \mathcal{S})$ where $\mathcal{S} = 1$ and, consequently, $m_{k+1} = m_0$ for every $k \ge 0$. It then follows from Lemma 2.2 with $\tilde{m} = m_0$ and $\nu = 1/m_0$ that $\overline{T}$ is bounded above by a unique integer $K \ge 1$ satisfying

$$\frac{K}{m_0} = \sum_{k=1}^{K}\frac{1}{m_{k+1}} \ge \frac{2\theta m_0 R_{1/m_0,m_0}(z_0)}{\rho^2} > \sum_{k=1}^{K-1}\frac{1}{m_{k+1}} = \frac{K-1}{m_0}, \tag{3.18}$$

where $R_{\tilde{m},\nu}(\cdot,\cdot)$ is as in (2.5). Using (3.18) and Lemma 3.6(c), we then have that

$$\overline{T} \le \sum_{k=0}^{K-1}\left\lceil \frac{4C_0}{\sqrt{m_{k+1}}}\right\rceil \le \sum_{k=0}^{K-1}\frac{8C_0}{\sqrt{m_{k+1}}} = \frac{8C_0K}{\sqrt{m_0}}$$

$$\overset{(3.18)}{<} \frac{8C_0}{\sqrt{m_0}}\left[1+\frac{2\theta m_0^2 R_{1/m_0,m_0}(z_0)}{\rho^2}\right]. \tag{3.19}$$

The bound in (3.16) now follows from (3.19) and (2.9) with $\nu = 1/m_0$ and $\tilde{m} = m_0$.□

Some comments about Algorithm 3.2 and Theorem 3.7 are in order. First, the total number of additional CF.ACG calls needed to find s on the over all calls of line (2) is at most $1+\log_1^+(\alpha m_*/m_0)$ when checking the ordered sequence of candidates $\{m_k\alpha^s\}_{s=0}^{\infty}$. Since this is on the same order of magnitude as in the quantities in (3.15)–(3.16), we do not count these additional inner iterations in our results. Second, Algorithm 3.2 shares some similarities with the smoothing/regularization method proposed by Nesterov in [28, Section 2.2.2]. The main differences are that we consider a nonsmooth composite term h in our objective function rather than fixing $h = 0$, there are (possibly) multiple outer iterations in CF.APD rather than just one, and our termination conditions for each outer iteration are relative to ϕ and the prox residual $\|z_k - z_{k+1}\|$ rather than relative to the tolerance ρ.

Before ending the section, we discuss how different choices of m_0 affect the complexities in (3.15) and (3.16) when $m_* \le M_*$. First, if $m_0 = \rho$ (resp. $m_0 = \rho^2$) then (3.16) and (3.15) are on the same order of complexity as in the bounds in the second and third columns of Table 1.1 for APD$\{\rho\}$ (resp. APD$\{\rho^2\}$) with $(m, M) = (m_*, M_*)$. Second, if d_0 (resp. Δ_0) is known, then choosing $m_0 = \rho/d_0$ (resp. $m_0 = \rho^2/\Delta_0$) implies that the bound in (3.16) is $\tilde{\mathcal{O}}(\sqrt{M_* d_0}/\sqrt{\rho})$ (resp. $\tilde{\mathcal{O}}(\sqrt{M_*\Delta_0}/\rho)$) which is optimal[3], up to logarithmic terms, for finding stationary points of (1) in the convex setting. Similarly, if f is nonconvex, then choosing $m_0 = 1$ implies that the bound in (3.15) is $\mathcal{O}(\sqrt{M_* m_*}\Delta_0/\rho^2)$ which matches the complexity of the AIPP

[3] See [28, Section 2.2.2] or [5, Theorem 1].

in [13] and is optimal[4] for finding stationary points of (1) in the weakly-convex setting. It is still unknown whether the logarithmic terms in the complexities above can be removed through a modification or extension of CF.APD.

4. Applications.

This section describes a few possible applications of Algorithm 3.2 in more general optimization frameworks.

Min-Max Smoothing. In [17], a smoothing framework was proposed for finding ε-stationary points of the nonconvex-concave min-max problem

$$\min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^l} [\phi(x,y)+h(x)] \tag{4.1}$$

where h is as in assumption ⟨A1⟩, $\phi(\cdot,y)$ is m_x-weakly convex and differentiable, $-\phi(x,\cdot)$ is proper closed convex with bounded domain, and $\nabla_x\phi(\cdot,\cdot)$ is Lipschitz continuous.

The framework specifically considers finding an ε-stationary point of a smooth approximation $\hat{p}$ of $\max_{y\in Y}\phi(\cdot,y)$ plus h. Choosing a special smoothing constant where the curvature pair $(\hat{m},\hat{M})$ of $\hat{p}$ satisfies $\hat{m}=m_x$ and $\hat{M}=\Theta(\varepsilon^{-1}D_y)$ (resp. $\hat{M}=\Theta(D_y^2\varepsilon^{-2})$), where D_y is diameter of $\mathrm{dom}(-\phi(x,\cdot))$, it was shown that an ε-stationary point of $\hat{p}$ yields an ε-primal-dual (resp. directional) stationary point of (4.1).

If we use CF.APD with $m_0=\rho$ to obtain an ε-stationary point of $\hat{p}$ as above, then an ε-primal-dual (resp. directional) stationary point of (4.1) is obtained in $\tilde{\mathcal{O}}(\varepsilon^{-2.5})$ (resp. $\tilde{\mathcal{O}}(\varepsilon^{-3})$) inner iterations which matches, up to logarithmic terms, the complexity bounds for the smoothing method in [17]. Moreover, when $\phi(\cdot,y)$ is convex, the above complexity is $\tilde{\mathcal{O}}(\varepsilon^{-1})$ (resp. $\tilde{\mathcal{O}}(\varepsilon^{-1.5})$), and this appears to be the first curvature-free approach that could be used for min-max optimization. This approach also has the particularly strong advantage that it does not need to know D_y.

Penalty Method. In [14], a penalty method is proposed for finding ε-KKT points of the linearly-constrained nonconvex optimization problem

$$\min_{x\in\mathbb{R}^n} \{\phi(x):=f(x)+h(x): Ax=b\} \tag{4.2}$$

where (f,h) are as in ⟨A1⟩–⟨A3⟩. It was shown that if the penalty method uses an algorithm $\mathcal{A}$ that needs $\mathcal{O}(T_{m,M}(\varepsilon))$ iterations to obtain an ε-stationary point of ϕ, then the total number inner iterations of the penalty method (for finding an ε-KKT point) is $\tilde{\mathcal{O}}(T_{m,\varepsilon^{-2}}(\varepsilon))$.

If we use the CF.APD with $m_0=\rho$ as algorithm $\mathcal{A}$ above, then an ε-KKT point of (4.2) is obtained in $\tilde{\mathcal{O}}(\varepsilon^{-3})$ inner iterations which matches the complexity bound for the particular penalty method in [14] (which uses the AIPP in [13] for algorithm $\mathcal{A}$). Moreover, when f is convex, the above complexity is $\tilde{\mathcal{O}}(\varepsilon^{-1.5})$. Like in the above discussion for min-max smoothing, this appears to be the first curvature-free approach used for linearly-constrained nonconvex optimization.

5. Numerical Experiments.

This section presents numerical experiments that support the practical viability of CF.APD.

We first describe the benchmark algorithms, the implementation of APD, the computing environment, common curvature estimates, and the problem of interest.

[4] See [33, Theorem 4.7].

The benchmark algorithms are instances of PGD, NCF, and UPF described in Section 1 and Table 1.1. Specifically, PGD chooses $\gamma_u = \gamma_d = 2$, NCF uses $\theta = 1.25$, and UPF uses $\gamma_1 = \gamma_2 = 0.4$, $\gamma_3 = 1$, $\beta_0 = 1$, and $\hat{\lambda}_0 = 1$. Moreover, UPF uses $\hat{\lambda}_k$ for the initial estimate of $\hat{\lambda}_{k+1}$ for $k \geq 1$. The implementations for NCF and UPF were generously provided by the respective authors of [20] and [9], while the PGD was implemented by the author.[5] Note that we did not consider the VAR-FISTA method in [32] because: (i) its steps were similar to NCF and (ii) we already had a readily available and optimized code for the NCF method.

The implementation of CF.APD, abbreviated as APD, is as described in Algorithm 3.2 with $\alpha = \beta = 2$ and the following additional updates at the beginning of every call to Algorithm 3.1 and the $(k+1)^{\text{th}}$ iteration of Algorithm 3.2, respectively:

$$L_0 \leftarrow \max\left\{L_0, \frac{L_0}{1+\beta/2}\right\}, \quad m_{k+1} \leftarrow \max\left\{m_0, \frac{m_{k+1}}{1+\alpha/2}\right\}.$$

This is done to allow a possible decrease in the curvature estimates. While we do not show convergence of this modified CF.APD, we believe that convergence can be established using similar techniques as in [27].

All experiments were run in MATLAB 2021a under a 64-bit Windows 10 machine with an Inter(R) Xeon(R) Gold 6240 processor and 12 GB of RAM. All benchmark algorithms use an initial curvature estimate of $(m_0, M_0) = (1, 1)$ while APD uses $(m_0, M_0) = (\rho, 1)$ following the insights in Subsection 3.2. A time limit of 2000 seconds was also prescribed.

The problem of interest is the 1225-variable nonconvex quadratic semidefinite programming (QSDP) problem:

$$\begin{aligned} (5.1) \qquad & \min_{Z \in \mathbb{R}^{35\times 35}} \; -\frac{\eta_1}{2}\|D\mathcal{B}(Z)\|^2 + \frac{\eta_2}{2}\|\mathcal{A}(Z) - b\|^2, \\ & \text{s.t. } \operatorname{tr}(Z) = 1, \quad Z \in \mathcal{S}^{35}_{+}, \end{aligned}$$

where $\mathcal{S}^n_+$ is the n-dimensional positive semidefinite cone, $\operatorname{tr}(Z)$ is the trace of a matrix, $b \in \mathbb{R}^{10}$, $D \in \mathbb{R}^{10\times 10}$ is a diagonal matrix with nonzero entries randomly generated from $\{1, ..., 1000\}$, $(\eta_1, \eta_2) \in \mathbb{R}^2_{++}$ are chosen to yield a particular curvature pair, and $\mathcal{A}, \mathcal{B} : \mathcal{S}^{35}_+ \mapsto \mathbb{R}^{10}$ are linear operators defined by

$$[\mathcal{A}(Z)]_j = A_j \bullet Z, \quad [\mathcal{B}(Z)]_j = B_j \bullet Z$$

for matrices $\{A_j\}_{j=1}^{10}, \{B_j\}_{j=1}^{10} \subseteq \mathbb{R}^{35\times 35}$. Moreover, the entries in these matrices and b were sampled from the uniform distribution on $[0, 1]$. The initial point z_0 given to each algorithm is the 35-dimensional identity matrix divided by 35. Each algorithm also tries to find a pair $(\bar{z}, \bar{v})$ solving Problem $\mathcal{CO}$ with f equal to the objective function of (5.1), h equal to the indicator function of the constraint set of (5.1), and $\rho = 10^{-5}(1 + \|\nabla f(z_0)\|)$.

Of the methods tested, PGD did not terminate with a stationary point in the prescribed time limit. The results of the experiment for the other methods are given in Table 5.1. Specifically, this table reports the number of evaluations of $(\partial h + \text{id})^{-1}$, i.e., the proximal operator of h, and runtime (in seconds) for different curvature pairs (m, M).

[5] See `https://github.com/wwkong/nc_opt/tree/master/tests/papers/apd` for the source code of the experiments.

		Iteration Count			Runtime		
m	M	UPF	NCF	APD	UPF	NCF	APD
5^1	5^3	18592	2133	**1664**	101.12	16.97	**8.42**
5^1	5^4	50778	10519	**5574**	290.19	89.08	**30.25**
5^1	5^5	71192	37643	**14610**	393.27	311.77	**73.36**
5^2	5^5	51968	31410	**6635**	287.02	257.06	**33.25**
5^3	5^5	19716	13880	**4921**	109.23	113.81	**25.51**
5^4	5^5	3364	2223	**643**	18.54	18.28	**3.21**

Table 5.1

Number of proximal evaluations of h and runtimes in the QSDP experiments for different curvature pairs (m, M). The bolded numbers indicate the best algorithm in terms of the number of evaluations (less is better) and runtime in seconds (less is better).

In all of the problem instances tested, CF.APD vastly outperformed its competitors (often by 3-4 times fewer iterations). This supports the immediate practical viability of CF.APD and opens the door to developing other computationally efficient variants of it.

6. Concluding Remarks.

This paper establishes iteration complexity bounds for CF.APD that are optimal, up to logarithmic terms, in terms of Δ_0 and d_0 (see (1.7)) when f is convex and in terms of Δ_0 when f is weakly-convex. It remains to be determined whether an optimal complexity bound in terms of d_0 exists for a curvature-free method when f is weakly convex.

In addition to the applications in Section 4, it would be interesting to see if CF.APD could be leveraged to develop a curvature-free proximal augmented Lagrangian method, following schemes similar to ones as in [15, 22].

Appendix A. Proof of Lemma 3.1.

The first result presents properties that are implied by the first inequality in (3.5).

Lemma A.1. *Given $(\mu, L) \in \mathbb{R}^2_{++}$ and $(y_j, x_j, A_j) \in \operatorname{dom}\psi^n \times \mathbb{R}^n \times \mathbb{R}_{++}$, let $(y_{j+1}, \tilde{x}_j, q^L_{j+1})$ be generated by (3.3) and (3.4). If, in addition, we have $\psi_s(y_{j+1}) - \ell_{\psi_s}(y_{j+1}; \tilde{x}_k) \le L\|y_{j+1} - \tilde{x}_j\|^2/2$, then:*

(a) $q^L_{j+1}(y_{j+1}) = \tilde{q}_{j+1}(y_{j+1})$ *and* $q^L_{j+1} \le \tilde{q}_{j+1}$;

(b) $\min\limits_{x\in\mathbb{R}^n} \left\{q^L_{j+1}(x) + L\|x - \tilde{x}_{j+1}\|^2/2\right\} = \min\limits_{x\in\mathbb{R}^n} \left\{\tilde{q}_{j+1}(x) + L\|x - \tilde{x}_{j+1}\|^2/2\right\}$;

(c) $x_{j+1} = \operatorname*{argmin}\limits_{x\in\mathbb{R}^n} \left\{a_j q^L_{j+1}(x)/[1 + \mu A_j] + \|x - x_j\|^2/2\right\}$;

(d) $x_{j+1} = \operatorname*{argmin}\limits_{x\in\mathbb{R}^n} \left\{A_{j+1} Q^L_{j+1}(x) + \|x - x_0\|^2/2\right\}$;

Proof. (a)–(b) See [12, Lemma B.0.1].

(c) Using the definition of q^L_{j+1}, the given optimality condition of x_{j+1} holds if and only if

$$x_{j+1} = x_j - \frac{a_j \nabla q^L_{j+1}(x_j)}{1 + \mu A_j} = x_j + \frac{a_j\left[L(y_{j+1} - \tilde{x}_j) + \mu(y_{j+1} - x_j)\right]}{1 + \mu A_j}$$

which is equivalent to the update for x_{j+1} in (3.3).

(d) See [12, Lemma B.0.3]. ∎

We are now ready to give the proof of Lemma 3.1.

Proof of Lemma 3.1. The first inequality of (3.5) follows immediately from (3.2) and our assumption that $L \geq L_*$. To show the second inequality of (3.5), we first derive two technical inequalities. For the first one, we use the fact that $a_j q^L_{j+1} + \xi_j\|\cdot\|^2/2$ is ξ_{j+1}-strongly convex, the definition of ξ_j, and the optimality of x_{j+1} in Lemma A.1(c) to obtain

$$a_j q^L_{j+1}(y_j) + \frac{\xi_j}{2}\|y_j - x_j\|^2 - \frac{\xi_{j+1}}{2}\|y_j - x_{j+1}\|^2 \geq a_j q^L_{j+1}(x_{j+1}) + \frac{\xi_j}{2}\|x_{j+1} - x_j\|^2. \tag{A.1}$$

For the second one, let $r_{j+1} = (A_j y_j + a_j x_{j+1})/A_{j+1}$. Using the convexity q^L_{j+1}, the fact that $a_j^2 = \xi_j A_{j+1}/L$ from (3.3), the definitions of $\tilde{x}_j$ and y_{j+1}, the fact that $A_{j+1} = a_j + A_j$, and Lemma A.1(b), we obtain

$$\begin{aligned}
&\frac{A_j q^L_{j+1}(y_j) + a_j q^L_{j+1}(x_{j+1})}{A_{j+1}} + \frac{\xi_j}{2A_{j+1}}\|x_{j+1} - x_j\|^2 \\
&\geq q^L_{j+1}(r_{j+1}) + \frac{\xi_j A_{j+1}}{2a_j^2}\left\| r_{j+1} - \frac{A_j y_j + a_j x_j}{A_{j+1}} \right\|^2 \\
&\overset{(3.3)}{=} q^L_{j+1}(r_{j+1}) + \frac{L}{2}\|r_{j+1} - \tilde{x}_j\|^2 \geq \min_{x\in\mathbb{R}^n}\left\{ q^L_{j+1}(x) + \frac{L}{2}\|x - \tilde{x}_j\|^2 \right\} \\
&\overset{\text{Lemma A.1(b)}}{=} \min_{x\in\mathbb{R}^n}\left\{ \tilde{q}_{j+1}(x) + \frac{L}{2}\|x - \tilde{x}_j\|^2 \right\} = \tilde{q}_{j+1}(y_{j+1}) + \frac{L}{2}\|y_{j+1} - \tilde{x}_j\|^2.
\end{aligned} \tag{A.2}$$

Combining (A.1), (A.2), (3.2), and our assumption that $L \geq L_*$, we conclude that

$$\begin{aligned}
&q^L_{j+1}(y_j) + \frac{\xi_j}{2A_{j+1}}\|y_j - x_j\|^2 - \frac{\xi_{j+1}}{2A_{j+1}}\|y_j - x_{j+1}\|^2 \\
&\overset{(A.1)}{\geq} \frac{A_j q^L_{j+1}(y_j) + a_j q^L_{j+1}(x_{j+1})}{A_{j+1}} + \frac{\xi_j}{2A_{j+1}}\|x_{j+1} - x_j\|^2 \\
&\overset{(A.2)}{\geq} \tilde{q}_{j+1}(y_{j+1}) + \frac{L}{2}\|y_{j+1} - \tilde{x}_j\|^2 \overset{(3.2)}{\geq} \psi(y_{j+1}) + \frac{L - L_* + \mu}{2}\|y_{j+1} - \tilde{x}_j\|^2 \\
&\geq \psi(y_{j+1}) + \frac{\mu}{2}\|y_{j+1} - \tilde{x}_j\|^2.
\end{aligned}$$

□

Appendix B. Proof of Lemma 3.3(d).

LEMMA B.1. *Let $\{(u_{j+1}, y_{j+1}, \tilde{x}_j, A_{j+1}, L_{j+1})\}_{j\geq 0}$ be generated by Algorithm 3.1 and suppose (3.6) holds for $L = L_{j+1}$ and every $j \geq 0$. Then, for every $j \geq 0$:*

(a) $A_{j+1}\psi(y_{j+1}) \leq \min_{x\in\mathbb{R}^n}\left\{A_{j+1}Q^L_{j+1}(x) + \|x - x_0\|^2/2\right\}$;

(b) $\xi_{j+1}\|x_{j+1} - y_{j+1}\|^2 \leq \|y_{j+1} - y_0\|^2$;

(c) if $\xi_j \geq 4$, then $\|y_{j+1} - \tilde{x}_j\|^2 \leq 13\|y_{j+1} - y_0\|^2/(\mu A_{j+1})$.

Proof. (a) See [12, Lemma B.0.3].

(b) Let $L = L_{j+1}$. Note that Q^L_{j+1} is a convex combination of $(j+1)$ μ-strongly convex quadratics. Hence, Q^L_{j+1} is a μ-strongly convex quadratic as well. Using Lemma A.1(d), part (a), the identity $x_0 = y_0$, and the previous fact we have that

$$\begin{aligned}
A_{j+1}\psi(y_{j+1}) &\overset{(a)}{\leq} \min_{x\in\mathbb{R}^n}\left\{ A_{j+1}Q^L_{j+1}(x) + \frac{1}{2}\|x - y_0\|^2 \right\} \\
&\overset{\text{Lemma A.1(d)}}{=} A_{j+1}Q^L_{j+1}(x_{j+1}) + \frac{1}{2}\|x_{j+1} - y_0\|^2 \\
&\leq A_{j+1}Q^L_{j+1}(y) + \frac{1}{2}\|y - y_0\|^2 - \frac{\xi_{j+1}}{2}\|y - x_{j+1}\|^2.
\end{aligned} \tag{B.1}$$

for any $y \in \mathbb{R}^n$. Re-arranging the above inequality at $y = y_{j+1}$ and using the second condition of (3.6), we conclude that

$$\frac{\|y_{j+1} - y_0\|^2 - \xi_{j+1}\|x_{j+1} - y_{j+1}\|^2}{2A_{j+1}} \overset{(B.1)}{\geq} \psi(y_{j+1}) - Q_j^L(y_{j+1}) \overset{(3.6)}{\geq} 0.$$

(c) Let $L = L_{j+1}$. Using (B.1) at $y = y_j$ and the second condition of (3.6), we first have that

$$\begin{aligned} A_{j+1}\left[\psi(y_{j+1}) - \psi(y_j)\right] &\overset{(3.6)}{\leq} A_{j+1}\left[\psi(y_{j+1}) - Q_{j+1}^L(y_j)\right] \\ &\overset{(B.1)}{\leq} \frac{1}{2}\|y_j - y_0\|^2 - \frac{\xi_{j+1}}{2}\|y_j - x_{j+1}\|^2 \end{aligned} \tag{B.2}$$

Combining this bound with the second condition in (3.5), the first condition in (3.6), and the fact that $\xi_j \leq \xi_{j+1}$, we have that

$$\begin{aligned} &\frac{\mu A_{j+1}}{2}\|y_{j+1} - \tilde{x}_j\|^2 + \frac{\xi_{j+1}}{2}\|y_j - x_{j+1}\|^2 \\ &\overset{(3.5)}{\leq} A_{j+1}\left[q_{j+1}^L(y_j) - \psi(y_{j+1})\right] + \frac{\xi_j}{2}\|y_j - x_j\|^2 \\ &\overset{(3.6)}{\leq} A_{j+1}\left[\psi(y_j) - \psi(y_{j+1})\right] + \frac{\xi_j}{2}\|y_j - x_j\|^2 \\ &\leq A_{j+1}\left[\psi(y_j) - \psi(y_{j+1})\right] + \frac{\xi_{j+1}}{2}\|y_j - x_j\|^2 \overset{(B.2)}{\leq} \frac{1}{2}\|y_j - y_0\|^2 \end{aligned} \tag{B.3}$$

We now bound $\|y_j - y_0\|$ in terms of $\|y_{j+1} - y_0\|$. Using the triangle inequality, (B.3), part (b), the fact that $\xi_j \leq \xi_{j+1}$, and our assumption that $\xi_j \geq 4$, it holds that

$$\begin{aligned} \|y_j - y_0\| &\leq \|y_0 - y_{j+1}\| + \|y_{j+1} - x_{j+1}\| + \|x_{j+1} - y_j\| \\ &\overset{(b)}{\leq} (1 + \xi_{j+1}^{-1})\|y_0 - y_{j+1}\| + \|x_{j+1} - y_j\| \\ &\overset{(B.3)}{\leq} (1 + \xi_{j+1}^{-1})\|y_0 - y_{j+1}\| + \sqrt{\frac{1}{\xi_{j+1}}}\|y_j - y_0\| \\ &\leq \frac{5}{4}\|y_0 - y_{j+1}\| + \frac{1}{2}\|y_j - y_0\|, \end{aligned}$$

and hence that $\|y_j - y_0\| \leq 5\|y_{j+1} - y_0\|/2$. Using this bound in (B.3), we conclude that

$$\|y_{j+1} - \tilde{x}_j\|^2 \overset{(B.3)}{\leq} \frac{2}{\mu A_{j+1}}\|y_j - y_0\|^2 \leq \frac{25}{2\mu A_{j+1}}\|y_{j+1} - y_0\|^2 \leq \frac{13}{\mu A_{j+1}}\|y_{j+1} - y_0\|^2.$$

□

We are now ready to give the proof of Lemma 3.3.

Proof of Lemma 3.3(d). Recall that we assumed ⟨A1⟩–⟨A2⟩ holds with $(f, g) = (\psi^s, \psi^n)$ and $\mathcal{M} = L_*$. Hence, it follows from Lemma 3.3(a), ⟨A2⟩ with $f = \psi^s$ and $\mathcal{M} = L_*$, and the triangle inequality, that

$$\begin{aligned} \|u_{j+1}\| &= \|(L_{j+1} + \mu)(\tilde{x}_j - y_{j+1}) + \nabla\psi^s(y_{j+1}) - \nabla\psi(\tilde{x}_j)\| \\ &\leq (L_{j+1} + \mu)\|y_{j+1} - \tilde{x}_j\| + \|\nabla\psi^s(y_{j+1}) - \nabla\psi(\tilde{x}_j)\| \\ &\leq 2L_{j+1}\|y_{j+1} - \tilde{x}_j\| + \|\nabla\psi^s(y_{j+1}) - \nabla\psi(\tilde{x}_j)\| \\ &\overset{\langle \mathtt{A2} \rangle}{\leq} (2L_{j+1} + L_*)\|y_{j+1} - \tilde{x}_j\|. \end{aligned}$$

Using the above bound, the fact that $L_{j+1} \geq L_0 \geq \mu$, and Lemma B.1(b), we conclude that

$$\|u_{j+1}\| \leq (2L_{j+1} + L_*)\|y_{j+1} - \tilde{x}_j\| \leq 3\bar{L}\|y_{j+1} - \tilde{x}_j\|$$

$$\overset{\text{Lemma B.1(b)}}{\leq} 3\bar{L}\sqrt{\frac{13}{\mu A_{j+1}}}\|y_{j+1} - y_0\| \leq \frac{12\bar{L}}{\sqrt{\mu A_{j+1}}}\|y_{j+1} - y_0\|. \qquad \square$$

REFERENCES

[1] M. M. Alves, R. D. C. Monteiro, and B. F. Svaiter, *Regularized HPE-type methods for solving monotone inclusions with improved pointwise iteration-complexity bounds*, SIAM Journal on Optimization, 26 (2016), pp. 2730–2743.

[2] A. Beck, *First-order methods in optimization*, SIAM, 2017.

[3] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.

[4] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, *Accelerated methods for nonconvex optimization*, SIAM Journal on Optimization, 28 (2018), pp. 1751–1772.

[5] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, *Lower bounds for finding stationary points II: first-order methods*, Mathematical Programming, 185 (2021), pp. 315–355.

[6] D. Drusvyatskiy and C. Paquette, *Efficiency of minimizing compositions of convex functions and smooth maps*, Mathematical Programming, 178 (2019), pp. 503–558.

[7] M. I. Florea and S. A. Vorobyov, *An accelerated composite gradient method for large-scale composite objective problems*, IEEE Transactions on Signal Processing, 67 (2018), pp. 444–459.

[8] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Mathematical Programming, 156 (2016), pp. 59–99.

[9] S. Ghadimi, G. Lan, and H. Zhang, *Generalized uniformly optimal methods for nonlinear programming*, Journal of Scientific Computing, 79 (2019), pp. 1854–1881.

[10] S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov, *On a combination of alternating minimization and Nesterov's momentum*, in International Conference on Machine Learning, PMLR, 2021, pp. 3886–3898.

[11] S. Guminov, Y. Nesterov, P. Dvurechensky, and A. Gasnikov, *Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems*, in Doklady Mathematics, vol. 99, Springer, 2019, pp. 125–128.

[12] W. Kong, *Accelerated inexact first-order methods for solving nonconvex composite optimization problems*, arXiv preprint arXiv:2104.09685, (2021).

[13] W. Kong, J. G. Melo, and R. D. C. Monteiro, *Complexity of a quadratic penalty accelerated inexact proximal point method for solving linearly constrained nonconvex composite programs*, SIAM Journal on Optimization, 29 (2019), pp. 2566–2593.

[14] W. Kong, J. G. Melo, and R. D. C. Monteiro, *An efficient adaptive accelerated inexact proximal point method for solving linearly constrained nonconvex composite problems*, Computational Optimization and Applications, 76 (2020), pp. 305–346.

[15] W. Kong, J. G. Melo, and R. D. C. Monteiro, *Iteration-complexity of a proximal augmented Lagrangian method for solving nonconvex composite optimization problems with nonlinear convex constraints*, arXiv preprint arXiv:2008.07080, (2020).

[16] W. Kong, J. G. Melo, and R. D. C. Monteiro, *FISTA and extensions-review and new insights*, arXiv preprint arXiv:2107.01267, (2021).

[17] W. Kong and R. D. C. Monteiro, *An accelerated inexact proximal point method for solving nonconvex-concave min-max problems*, SIAM Journal on Optimization, 31 (2021), pp. 2558–2585.

[18] H. Li and Z. Lin, *Accelerated proximal gradient methods for nonconvex programming*, Advances in neural information processing systems, 28 (2015).

[19] J. Liang and R. D. C. Monteiro, *A doubly accelerated inexact proximal point method for nonconvex composite optimization problems*, arXiv preprint arXiv:1811.11378, (2018).

[20] J. Liang, R. D. C. Monteiro, and C.-K. Sim, *A FISTA-type accelerated gradient algorithm for solving smooth nonconvex composite optimization problems*, Computational Optimization and Applications, 79 (2021), pp. 649–679.

[21] M. Marques Alves, R. D. C. Monteiro, and B. F. Svaiter, *Iteration-complexity of a Rockafellar's proximal method of multipliers for convex programming based on second-order*

approximations, Optimization, 68 (2019), pp. 1521–1550.

[22] J. G. Melo, R. D. C. Monteiro, and W. Kong, *Iteration-complexity of an inner accelerated inexact proximal augmented Lagrangian method based on the classical lagrangian function and a full Lagrange multiplier update*, arXiv preprint arXiv:2008.00562, (2020).

[23] R. D. C. Monteiro, C. Ortiz, and B. F. Svaiter, *An adaptive accelerated first-order method for convex optimization*, Computational Optimization and Applications, 64 (2016), pp. 31–73.

[24] R. D. C. Monteiro, M. R. Sicre, and B. F. Svaiter, *A hybrid proximal extragradient self-concordant primal barrier method for monotone variational inequalities*, SIAM Journal on Optimization, 25 (2015), pp. 1965–1996.

[25] R. D. C. Monteiro and B. F. Svaiter, *Convergence rate of inexact proximal point methods with relative error criteria for convex optimization*, submitted to SIAM Journal on Optimization, (2010).

[26] R. D. C. Monteiro and B. F. Svaiter, *On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean*, SIAM Journal on Optimization, 20 (2010), pp. 2755–2787.

[27] Y. Nesterov, *Gradient methods for minimizing composite functions*, Mathematical programming, 140 (2013), pp. 125–161.

[28] Y. Nesterov, *Lectures on convex optimization*, vol. 137, Springer, 2 ed., 2018.

[29] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky, *Primal–dual accelerated gradient methods with small-dimensional relaxation oracle*, Optimization Methods and Software, (2020), pp. 1–38.

[30] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui, *Catalyst acceleration for gradient-based non-convex optimization*, arXiv preprint arXiv:1703.10993, (2017).

[31] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of operations research, 1 (1976), pp. 97–116.

[32] C.-K. Sim, *A FISTA-type first order algorithm on composite optimization problems that is adaptable to the convex situation*, arXiv preprint arXiv:2008.09911, (2020).

[33] D. Zhou and Q. Gu, *Lower bounds for smooth nonconvex finite-sum optimization*, in International Conference on Machine Learning, PMLR, 2019, pp. 7574–7583.